Please amend the following claims:

1. (three times amended) A method for removing an overflow condition comprising the steps of:

detecting a first digitally encoded data stream portion causing said overflow condition;

delaying said first data stream portion for a delay time that prevents said overflow condition; and

accelerating a second data stream portion that follows said first data stream portion to substantially make-up for said delay time.

2. (three times amended) An apparatus for removing an overflow condition comprising:

means for detecting a first digitally encoded data stream portion causing an overflow condition;

means for delaying said first data stream portion; and

means for accelerating a second data stream portion that follows said first data stream portion.

4. (twice amended) A method for avoiding overflow of a decoder buffer comprising:

(a) determining a total amount of old data stream data that, if transmitted to said decoder buffer, would occupy said decoder buffer;

(b) adding, to said total amount, an amount of new data stream data to obtain a combined amount of data;

(c) testing if said combined amount of data would overflow said decoder buffer; and

(d) if overflow would occur, then causing a portion of the new data stream to be delayed by a delay amount corresponding to at least said overflow, if said portion of the new data stream were to be transmitted to said decoder buffer.

5. (twice amended) The method according to claim 4, wherein the step (a) of determining is preceded by determining a maximum size of said decoder buffer.

6. (amended) The method according to claim 5, wherein said maximum size is determined

according to a buffer size parameter within the old data stream.

7. (amended) The method according to claim 5, wherein said maximum size of step (a) is determined according to a buffer size parameter within the new data stream.

8. (twice amended) The method according to claim 4, further comprising:

prior to testing of step (c), subtracting, from said total amount, an amount of old data stream data that, if transmitted, would be decoded by a decoder;

9. (twice amended) The method according to claim 4, wherein said delay amount of step (d) is a function of an amount of data stream data by which said decoder buffer is overflowed within said portion of the new data stream.

10. (twice amended) The method according to claim 4, wherein said delay amount of step (d) is a function of an amount of data stream data by which said decoder buffer is overflowed in a single instance of overflow within said portion of the new data stream.

---

17. (twice amended) The method according to claim 16, wherein said second plurality of old data stream frame sizes of step (d) include all frames of the old data stream portion that will remain un-decoded when said new data stream frame will be received by the decoder, if the data stream portions are transmitted.

18. (amended) The method according to claim 17, wherein the data stream portions are transmitted.

19. (twice amended) The method according to claim 16, further comprising:

if overflow is found in step (f), then causing a transmission time of a portion of new data stream data including said new data stream frame to be delayed.

---

c4

21. (amended) The method according to claim 20, wherein said delay is caused by adding null packets to said new data stream portion.

c5

23. (amended) The method according to claim 22, wherein m equals 1504.

25. (amended) The method according to claim 24, wherein said modified new data stream timing reference of step (a) further corresponds with a timing gap between a first decoding time for decoding a last frame of the old data stream and a second decoding time for decoding a first frame of the new data stream.

26. (amended) The method according to claim 24, wherein determining said modified new data stream timing reference includes:

(i) determining said current timing reference of the new data stream;

(ii) determining a delay between said current timing reference and a current decoding time of a frame of the new data stream;

c6

(iii) determining a new decoding time of said frame of the new data stream that corresponds with a sum of said current decoding time and an inter-frame delay between a decoding time for decoding a last frame of the old data stream and a decoding time for decoding a first frame of the new data stream; and

(iv) determining said modified new data stream timing reference as said new decoding time of step (iii) minus said delay of step (ii).

27. (twice amended) The method according to claim 24, wherein determining said modified new data stream timing reference includes:

(i) determining a program clock reference of a first packet of said new data stream;

(ii) determining a delay between transmission of a first sequence header of said new data stream and a first decode time stamp DTS of a first frame of said new data stream;

(iii) determining a continuous DTS as a sum of said first DTS and an inter-frame delay; and

(iv) determining a new data stream real-time transmit time as said continuous DTS of step (iii) minus said delay of step (ii).

28. (twice amended) The method according to claim 24, wherein said aligning in step (b) sets a start time for transmitting the portion of the new data stream that corresponds with a decoding time for decoding the portion of the old data stream.

29. (twice amended) The method according to claim 24, wherein said aligning in step (b) sets a start time for a decoder buffer to begin receiving the portion of the new data stream that corresponds with a decoding time for decoding the portion of the old data stream.

30. (amended) The method according to claim 24, further comprising:

(d) detecting a decoder buffer overflow condition that will result from said splicing, if the data streams are transmitted; and

(e) correcting said overflow condition.

31. (twice amended) The method according to claim 24, wherein said determining of step (a) is preceded by

(i) determining the splice-out point of the old data stream; and

(ii) determining the splice-in point of the new data stream.

32. (amended) The method according to claim 31, wherein step (ii) includes, if an initial frame of the new data stream is of a type that is ordinarily decoded with reference to decoding of a prior frame, then modifying the new data stream to remove said reference.

33. (twice amended) The method according to claim 32, wherein said frame type is selected from a group consisting of B-frames and P-frames, and wherein said step of modifying comprises closing an open group of pictures GOP.

34. (amended) The method according to claim 31, wherein said data streams include video and audio data, wherein step (a) includes determining a video splice-out point and an audio splice-out point, and wherein step (b) includes determining a video splice-in point and an audio splice-in point.

35. (amended) The method according to claim 31, wherein said splice-out point of step (i) is determined within a user-selectable portion of the old data stream.

36. (amended) The method according to claim 31, wherein said splice-in point of step (ii) is determined within a user-selectable portion of the new data stream.

37. (amended) The method according to claim 31, wherein said splice-out point of step (i) is user-selectable.

38. (amended) The method according to claim 31, wherein said splice-in point of step (ii) is user-selectable.

39. (amended) The method according to claim 24, wherein step (a) is preceded by determining a first source for the old data stream and a second source for the new data stream.

40. (twice amended) The method according to claim 39, wherein said first and second sources include source types selected from a group comprising a storage device, a satellite receiver, a

cable receiver, a network, an audio source, a video source and an encoder.

41. (amended) The method according to claim 40, wherein said first source and said second source are of a same source type.

42. (amended) The method according to claim 24, wherein at least one of said data streams is MPEG encoded.

c6 cont'd.

43. (amended) The method according to claim 24, wherein said splicing is accomplished in real-time.

44. (amended) The method according to claim 24, wherein step (a) is followed by transmitting a portion of the old data stream.

45. (twice amended) The method according to claim 24, wherein step (b) is followed by transmitting the portion of the new data stream.

---

49. (twice amended) The computer-readable storage medium according to claim 47, wherein the step of determining the splice-out point is preceded by:

creating at least one data storage structure for storing portions of said old and new data streams; and

c7

storing portions of said old and new data streams in said at least one data storage structure.

50. (amended) The computer readable medium according to claim 49, wherein said at least one data storage location is located in memory of a host processing system.

---

52. (twice amended) The method according to claim 51, wherein said dependency of step (f) is an open GOP and wherein said modifying closes the open GOP.

c8

53. (twice amended) The method according to claim 51, further comprising:

checking for overflow of said decoder buffer; and

if overflow is found, then removing said overflow.

55. (twice amended) A method for preparing a digitally encoded data stream for splicing, comprising:

(a) determining a splice-in point of a new data stream; and

(b) closing an initial open group of pictures GOP of the new data stream, if the new data stream includes an initial open GOP, wherein the step of closing the GOP includes the step of removing open frames.

56. (twice amended) A splicer for splicing digitally encoded data streams including an old data stream and a new data stream, comprising:

c9

(a) means for determining a splice-in point of the new data stream; and

(b) means for closing an open group of pictures GOP of the new data stream, if the new data stream includes the open GOP, wherein the step of closing the GOP includes the step of removing open frames.

57. (amended) The method according to claim 51, wherein said splice-out point is determined in step (a) according to a user selection between an insert mode option and a splice-only mode option.

58. (twice amended) The method according to claim 57, wherein said splice-out point is determined as being immediately prior to a sequence header.

59. (twice amended) The method according to claim 57, wherein said splice-out point is determined as being immediately prior to a first occurring one of a group of pictures GOP header, an

I-frame and a P-frame.

60. (twice amended) The method according to claim 51, wherein said step of determining said splice-in point comprises:

finding a decode time stamp ("DTS") for a frame of the new data stream, said frame being included within a group of pictures GOP of the new data stream;

finding a corresponding presentation time stamp for said frame of the new data stream; and

if said frame of the new data stream is other than an I-frame, then closing said GOP.

C9 Cont'd

61. (twice amended) The method according to claim 60, wherein said frame is the initial frame of the new data stream.

62. (amended) The method according to claim 60, wherein finding said DTS includes parsing a first portion of the new data stream for a first sequence header, and then further parsing said first portion for a last DTS before a first frame header.

C10

64. (twice amended) The method according to claim 63, wherein step (b) is accomplished by deleting another frame within said portion that precedes said independently decodable frame.

C11

66. (amended) The method according to claim 65, wherein step (d) of modifying includes replacing increasing temporal reference values of remaining frames within said GOP with correspondingly increasing temporal reference values of frames deleted in step (c).

C12

69. (amended) The method according to claim 68 wherein said step of finding is accomplished in real-time during splicing of said new data stream with said old data stream.

70. (amended) The method according to claim 68 wherein said inter-frame delay equals 1001/30,000 seconds.

71. (twice amended) A method for aligning a splice-out portion of a digitally encoded old data stream with a splice-in portion of a digitally encoded new data stream, comprising the step of setting a start of receipt time of said new data stream at which, if said new data stream is transmitted, then said new data stream will begin to be received by a decoder in alignment with a decoding time for said splice-out portion of said old data stream, and wherein said step of setting includes:

if, upon transmission of said old and new data streams, said new data stream would begin to be received by a decoder before the decoder would have received all of said splice-out portion, then setting a transmission delay parameter for said new data stream; and

if said new data stream, upon transmission, would begin to be received by the decoder after the decoder has received all of said splice-out portion, then setting a transmission acceleration parameter for said new data stream.

C12 Cont.

73. (amended) The method according to claim 71 that further includes inserting a number of null packets corresponding with said delay parameter into said new data stream at a position such that said null packets will be transmitted substantially prior to other new data stream data, if said new data stream is transmitted.

74. (twice amended) The method according to claim 73 wherein said number of null packets equals a number of data packets that, without inserting the null packets, would be received by the decoder before the decoder has received all of said splice-out portion, if the new data stream is transmitted.

76. (Twice amended) The method according to claim 71 that further includes deleting a number of null packets corresponding with said acceleration parameter from a first transmitted portion of said new data stream, if said new data stream is transmitted.

77. (twice amended) The method according to claim 76 wherein said number of null packets equals a number of data packets that, without said deleting, would be received by the decoder after

C13 Cont'd:

the decoder has received all of said splice-out portion, if the new data stream is transmitted.

80. (twice amended) The apparatus according to claim 79, wherein said data stream portions include a new data stream portion with new data stream data and said new data stream data is received as a plurality of data packets.

C13

81. (amended) The apparatus according to claim 80, wherein said amount is calculated as a time corresponding to a number of data packets of data stream data.

83. (amended) The method according to claim 51 wherein the step of modifying the portion of the old data stream includes the step of dropping packets that exist in the old data stream.

Please add the following claims:

84. The method according to claim 1 wherein the step of delaying inserts null packets and the step of accelerating deletes other null packets.

85. The method according to claim 1 wherein the steps of detecting, delaying and accelerating are performed in real-time.

C14

86. The method according to claim 85 wherein special splicing characters are not recognized.

87. The method according to claim 85 wherein the step of delaying inserts null packets and the step of accelerating deletes other null packets.

88. The method according to claim 20 wherein causing the delay inserts null packets and causing the acceleration deletes other null packets.

89. The method according to claim 88 wherein the delay inserts null packets at a beginning of the new data stream portion.

90. The method according to claim 88 wherein an amount of the inserted null packets corresponds to another amount of deleted other null packets.

91. The method according to claim 20 wherein the causing the delay and the causing the acceleration are performed in real-time.

92. The method according to claim 91 wherein special splicing characters are not recognized.

93. The method according to claim 91 wherein causing the delay inserts null packets and causing the acceleration deletes other null packets.

94. The splicer according to claim 54 wherein the means for aligning causes the delaying by inserting null packets and causes the accelerating by deleting other null packets.

95. The splicer according to claim 94 wherein the means for aligning deletes the other null packets in an amount corresponding to a number of inserted null packets.

C14 cont.

96. The apparatus according to claim 79 wherein the shifting means determines the amount of delayed transmission times by determining insertion points for null packets and determines the amount of accelerated transmission times by determining deletion points for other null packets.

97. The apparatus according to claim 96 wherein the shifting means determines that the amount of delayed transmission times corresponds to the amount of accelerated transmission times.

98. A method for removing an overflow condition comprising the steps of:

obtaining first and second digitally encoded data stream portions, with at least one of the first and second digitally encoded data stream portions containing no special splicing characters;

outputting the first digitally encoded data stream portion;

detecting whether a portion of the output of the second digitally encoded data stream portion would cause said overflow condition;

delaying a part of said second data stream portion for a delay time that prevents said overflow

condition; and

accelerating a subsequent portion of the second data stream that follows the portion of the second data stream portion to substantially make-up for said delay time.

99. The method according to claim 98 wherein the step of delaying inserts null packets and the step of accelerating deletes other null packets.

100. The method according to claim 99 wherein the delay inserts null packets at a beginning of the new data stream portion.

101. The method according to claim 99 wherein an amount of the inserted null packets corresponds to another amount of deleted other null packets.

102. The method according to claim 98 wherein the steps of delaying and accelerating are performed in real-time.

103. The method according to claim 102 wherein the step of delaying inserts null packets and the step of accelerating deletes other null packets.

104. The method according to claim 103 wherein the delay inserts null packets at a beginning of the new data stream portion.

105. The method according to claim 103 wherein an amount of the inserted null packets corresponds to another amount of deleted other null packets.